

日本国特許庁
JAPAN PATENT OFFICE

Hiroataka YOSHIDA et al

McDermott Will & Emery LLP

別紙添付の書類に記載されている事項は下記の出願書類に記載されて
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
with this Office.

出願年月日 2003年 6月 3日
Date of Application:

出願番号 特願2003-157444
Application Number:
[ST. 10/C]: [JP 2003-157444]

願人 株式会社日立製作所
Applicant(s):

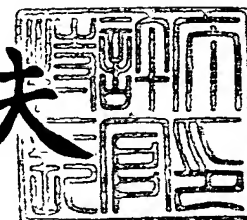
CERTIFIED COPY OF
PRIORITY DOCUMENT

BEST AVAILABLE COPY

2004年 2月25日

特許庁長官
Commissioner,
Japan Patent Office

今井康夫



出証番号 出証特2004-3013346

【書類名】 特許願

【整理番号】 K03003151A

【あて先】 特許庁長官殿

【国際特許分類】 G09C 1/00

【発明者】

 【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所システム開発研究所内

 【氏名】 吉田 博隆

【発明者】

 【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所システム開発研究所内

 【氏名】 古屋 聡一

【特許出願人】

 【識別番号】 000005108

 【氏名又は名称】 株式会社 日立製作所

【代理人】

 【識別番号】 100075096

 【弁理士】

 【氏名又は名称】 作田 康夫

【手数料の表示】

 【予納台帳番号】 013088

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

 【物件名】 図面 1

 【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 改竄検知可能な、共通鍵暗号の暗号化装置または復号化装置

【特許請求の範囲】

【請求項 1】

共通鍵暗号の暗号化装置であって、

冗長データとメッセージとからなる平文を特定の長さで区切った複数の平文ブロックを生成する処理部と、

秘密鍵から、前記平文よりも長い乱数列を生成し、前記乱数列から前記平文ブロックに対応する暗号化のための乱数ブロックを生成し、前記平文ブロックと前記乱数ブロックを用いて暗号文ブロックを暗号演算する処理部と、

前記乱数列から前記暗号文ブロックに対応する認証のための乱数ブロックを生成し、暗号文ブロックと前記乱数ブロックを用いてメッセージ認証子ブロックを認証演算する処理部とを備える共通鍵暗号の暗号化装置。

【請求項 2】

請求項1記載の共通鍵暗号の暗号化装置であって、

前記暗号演算処理部と前記認証演算処理部は一つ以上の、その合計の長さは、前記平文ブロックの合計より長く平文ブロックの合計の2倍より短い前記乱数ブロックを用いる共通鍵暗号の暗号化装置。

【請求項 3】

請求項 2 記載の共通鍵暗号の暗号化装置であって、

前記暗号演算処理部は、前記平文ブロックを用いた2項演算や単項演算を決められた手順に従って1回以上行い、

前記認証演算処理部は、前記暗号文ブロックを用いた2項演算や単項演算を決められた手順に従って1回以上行い、

得られた複数の暗号文ブロックとメッセージ認証子ブロックを組み合わせ、暗号文として出力する処理部を備える共通鍵暗号の暗号化装置。

【請求項 4】

請求項 2 記載の共通鍵暗号の暗号化装置であって、

前記暗号演算処理部は、前記暗号演算を排他的論理和によって行い、

前記認証演算処理部は、前記認証演算を算術乗算と算術加算により行う共通鍵暗号の暗号化装置。

【請求項 5】

請求項 2 記載の共通鍵暗号の暗号化装置であって、
前記暗号演算処理部は、前記暗号演算を排他的論理和によって行い、
前記認証演算処理部は、前記認証演算を有限体上の乗算と算術加算により行う共通鍵暗号の暗号化装置。

【請求項 6】

請求項 2 記載の共通鍵暗号の暗号化装置であって、
前記暗号演算処理部と前記認証演算処理部とは、使う乱数ブロックを共有する共通鍵暗号の暗号化装置。

【請求項 7】

請求項 2 記載の共通鍵暗号の暗号化装置であって、
前記暗号演算処理部と前記認証演算処理部とは、使う乱数ブロックを共有する共通鍵暗号の暗号化装置。

【請求項 8】

請求項 2 記載の共通鍵暗号の暗号化装置であって、
前記暗号演算処理部と前記認証演算処理部とは、異なる乱数ブロックを使用する共通鍵暗号の暗号化装置。

【請求項 9】

請求項 2 記載の共通鍵暗号の暗号化装置であって、
前記秘密鍵から前記乱数列を生成する擬似乱数生成処理部を備える共通鍵暗号の暗号化装置。

【請求項 1 0】

請求項 9 の共通鍵暗号の暗号化装置であって、
前記メッセージを複数に分割する処理部を備え、
前記擬似乱数生成処理部は、前記分割数分の乱数列を生成し、
前記分割したメッセージと前記乱数列のいずれかをそれぞれ異なる演算ユニットに割り当て、並列処理を行わせる処理部を備える共通鍵暗号の暗号化装置。

【請求項 1 1】

共通鍵暗号の復号化装置であって、
暗号文を特定の長さで区切った複数の暗号文ブロックを生成する処理部と、
秘密鍵から、前記暗号文よりも長い乱数列を生成し、前記乱数列から前記暗号文ブロックに対応する認証のための乱数ブロックを生成し、暗号文ブロックと前記乱数ブロックを用いてメッセージ認証子ブロックを認証演算する処理部と、
前記乱数列から暗号文ブロックに対応する復号のための乱数ブロックを生成し、前記暗号文ブロックと、前記乱数ブロックを用いて平文ブロックを復号演算する処理部とを備える共通鍵暗号の復号化装置。

【請求項 1 2】

請求項 1 1 記載の共通鍵暗号の復号化装置であって、
前記認証演算処理部と前記復号演算処理部は一つ以上の、その合計の長さは、前記平文ブロックの合計より長く平文ブロックの合計の2倍より短い前記乱数ブロックを用いる共通鍵暗号の復号化装置。

【請求項 1 3】

請求項 1 2 記載の共通鍵暗号の復号化装置であって、
前記平文ブロックを複数連結して平文を生成する処理部と、
前記平文に含まれる冗長データを抽出する処理部と、
前記冗長データを検査し、前記暗号文への改ざんの有無を検出する処理部とを備える共通鍵暗号の復号化装置。

【請求項 1 4】

コンピュータに、共通鍵暗号の暗号化処理を実行させるプログラムを記憶した媒体であって、
前記プログラムは、前記コンピュータに、
冗長データとメッセージとからなる平文を特定の長さで区切った複数の平文ブロックを生成させ、
秘密鍵から、前記平文よりも長い乱数列を生成し、前記乱数列から前記平文ブロックに対応する暗号化のための乱数ブロックを生成し、前記平文ブロックと前記乱数ブロックを用いて暗号文ブロックを暗号演算させ、

前記乱数列から前記暗号文ブロックに対応する認証のための乱数ブロックを生成させ、暗号文ブロックと前記乱数ブロックを用いてメッセージ認証子ブロックを認証演算させる、プログラムを記憶した媒体。

【請求項 1 5】

請求項 1 4 のプログラムを記憶した媒体であって、

前記暗号演算と前記認証演算は一つ以上の、その合計の長さは、前記平文ブロックの合計より長く平文ブロックの合計の2倍より短い前記乱数ブロックを用いる、プログラムを記憶した媒体。

【請求項 1 6】

請求項 1 5 のプログラムを記憶した媒体であって、

前記暗号演算として、前記平文ブロックを用いた2項演算や単項演算を決められた手順に従って1回以上行わせ、

前記認証演算として、前記暗号文ブロックを用いた2項演算や単項演算を決められた手順に従って1回以上行わせ、

得られた複数の暗号文ブロックとメッセージ認証子ブロックを組み合わせ、暗号文として出力させる、プログラムを記憶した媒体。

【請求項 1 7】

請求項 1 5 のプログラムを記憶した媒体であって、

前記暗号演算を排他的論理和によって行わせ、

前記認証演算を算術乗算と算術加算により行わせる、プログラムを記憶した媒体。

【請求項 1 8】

請求項 1 5 のプログラムを記憶した媒体であって、

前記暗号演算を排他的論理和によって行わせ、

前記認証演算を有限体上の乗算と算術加算により行わせる、プログラムを記憶した媒体。

【請求項 1 9】

請求項 1 5 のプログラムを記憶した媒体であって、

前記暗号演算と前記認証演算とが使う乱数ブロックを共有させる、プログラム

を記憶した媒体。

【請求項 20】

請求項 15 のプログラムを記憶した媒体であって、
前記暗号演算と前記認証演算とが使う乱数ブロックを共有させる、プログラムを記憶した媒体。

【請求項 21】

請求項 15 のプログラムを記憶した媒体であって、
前記秘密鍵から前記乱数列を生成させる擬似乱数生成処理を行わせる、プログラムを記憶した媒体。

【請求項 22】

請求項 21 のプログラムを記憶した媒体であって、
前記メッセージを複数に分割し、
前記擬似乱数生成処理によって、前記分割数分の乱数列を生成させ、
前記分割したメッセージと前記乱数列のいずれかをそれぞれ異なる演算ユニットに割り当て、並列処理を行わせる、プログラムを記憶した媒体。

【請求項 23】

コンピュータに、共通鍵暗号の復号化処理を実行させるプログラムを記憶した媒体であって、
前記プログラムは、前記コンピュータに、
暗号文を特定の長さで区切った複数の暗号文ブロックを生成させ、
秘密鍵から、前記暗号文よりも長い乱数列を生成させ、前記乱数列から前記暗号文ブロックに対応する認証のための乱数ブロックを生成させ、暗号文ブロックと前記乱数ブロックを用いてメッセージ認証子ブロックを認証演算させ、
前記乱数列から暗号文ブロックに対応する復号のための乱数ブロックを生成させ、前記暗号文ブロックと、前記乱数ブロックを用いて平文ブロックを復号演算させる、プログラムを記憶した媒体。

【請求項 24】

請求項 23 のプログラムを記憶した媒体であって、
前記復号演算と前記認証演算とに一つ以上の、その合計の長さは、前記平文ブ

ロックの合計より長く平文ブロックの合計の2倍より短い前記乱数ブロックを用いさせる、プログラムを記憶した媒体。

【請求項 2 5】

請求項 2 4 のプログラムを記憶した媒体であって、
前記平文ブロックを複数連結して平文を生成させ、
前記平文に含まれる冗長データを抽出させ、
前記冗長データを検査させ、前記暗号文への改ざんの有無を検出させる、プログラムを記憶した媒体。

【発明の詳細な説明】

【 0 0 0 1 】

【発明の属する技術分野】

本発明は、秘密情報のセキュリティを確保する技術に関する。

【 0 0 0 2 】

【従来の技術】

従来の暗号処理装置は、データを秘匿する目的のブロック暗号やストリーム暗号が提案されていた。ブロック暗号にはAES (Advanced Encryption Standard) を始め、いろいろなアルゴリズムが提案されている。

【 0 0 0 3 】

ブロック暗号はECB, CBC, CFB, OFB, カウンタモードなどのブロック暗号操作モードにより全体の暗号処理の安全性や性質を議論するが、これまでに暗号化処理と改ざん検出を同時に行う操作モードは、iaPCBCモードが知られているだけで、残りのモードは改ざんの検出がそれ自身では不可能である。iaPCBCモードは非特許文献 1 で扱っている。

【 0 0 0 4 】

iaPCBCモードはブロック暗号を用いた操作モードであり、暗号化処理では並列処理、事前計算などができないので、高速な処理が要求される環境への実装が困難であった。

【 0 0 0 5 】

これに対して、メッセージ認証子と呼ばれる改ざん検出のための暗号学的チェ

ックサム(以下、MACという)を生成する方法が提案され、ブロック暗号の上記操作モードの暗号処理でも必要に応じてMAC生成処理を同時にかつまったく独立の機構として実装することで、暗号処理と改ざん検出が同時に可能となった。しかし、この場合にはまったく独立な暗号学的鍵を2度、すなわち、暗号化用と改ざん検出用、共有する必要があるという点、それから暗号化されるデータを2度処理、つまり、暗号化処理とMAC生成処理、にかけの必要があり、システムが複雑になったり、長いデータの処理に向かなくなったりなどの懸念があった。さらにブロック暗号の処理速度が現在の通信の速度に比べて低速であり、これらブロック暗号とMACの組み合わせ技術は、ギガビットやテラビット処理といった高速処理が要求される用途への応用が困難であった。

【0 0 0 6】

MACと軽い処理との組み合わせが操作モードを実現できることが知られていた。それをモードとして用いたストリーム暗号は、暗号処理と改ざん検出を同時に行えるほか、上記ブロック暗号の処理にくらべて、2倍～20倍の比率でストリーム暗号がより高速であるが、どのMAC生成方法についても、ブロック暗号とMACの組み合わせ同様、メッセージの長さに対してその2倍の長さの擬似乱数を必要とし、必要な乱数生成に時間がかかったり、ひとつのメッセージについて2度の処理を行う必要などがあった。

【0 0 0 7】

より詳しくMAC生成方法を考えると、本来のストリーム暗号に、付帯的に必要となるメカニズムや計算量が非常に大きい。例えば、UMACなどのMAC生成方法では暗号学的に衝突なしで一方向性を保証している安全なハッシュ関数が必要であり、ストリーム暗号として用いるには、擬似乱数器にさらに前述のハッシュ関数を実装する必要がある。、UMACは以下の非特許文献2で扱っている。

【0 0 0 8】

【非特許文献1】

V.Gligor, P.Donescu著「Lecture notes in Computer Science, vl.1796」Springer-Verlag出版, 2000年, p.153-171

【非特許文献2】

Black, Halevi, Krawczyk, Krovetz, Rogaway, “UMAC: Fast and Secure Message Authentication,” Advances in Cryptology, - CRYPTO’ 99, Lecture Notes in Computer Science, Vol. 1666, Springer-Verlag, 1999

【0 0 0 9】

【発明が解決しようとする課題】

従来の暗号技術のほとんどは、復号化の際、改ざん検出をそれ自身ではすることができなかった。改ざん検出を行うとき、異なるふたつの鍵共有の必要性、メッセージの2倍となる乱数の必要性、独立の処理、別の暗号学的要素関数の追加実装などが必要であった。

【0 0 1 0】

処理速度の面についての課題として、ブロック暗号のこれまで知られた操作モードでは並列度や事前計算等の可能性がなく、高並列処理や高速処理には向いていないという点があり、ストリーム暗号のこれまで知られた操作モードでは、演算量が多いということや、必要な乱数が多いという理由で、ソフトウェア実装における処理速度がブロック暗号の場合と同程度であり、より高速な処理速度が必要とされている。

【0 0 1 1】

【課題を解決するための手段】

本発明は効率的で証明可能安全な暗号方法であり、改ざん検出も復号化と同時に可能であり、またデータ秘匿、データ改ざんに対する安全性については、証明可能なメッセージ認証付き暗号方法とその装置を提供する。

【0 0 1 2】

本発明は擬似乱数生成器の高速処理性能を生かしながら、事前計算や並列処理の利点をもつ、共通鍵暗号の方法とその装置を提供する。

【0 0 1 3】

本発明は従来のブロック暗号よりも高速な処理が可能というだけでなく、シングルパスで実装が可能でソフトウェアにおいて極めて効率的な処理が可能な暗号方法とその装置を提供する。

【0 0 1 4】

本発明は小さなプログラムで実装可能なストリーム暗号方法とその装置を提供する。

【0015】

本発明は、その一態様において、乱数を生成して暗号化と認証処理を行い、事前計算と並列計算を達成する。また、生成する乱数の長さは、メッセージ長Nに対して $2N$ より少ない乱数を用いて、暗号処理と認証処理を行う。

【0016】

具体的には、疑似乱数生成器を使って乱数を生成しそれらをブロック毎に分割する。また平文もブロック毎に分割する。乱数ブロックと平文ブロックを排他的論理和し暗号文ブロックを得る。非特許文献2で扱っているハッシュ関数NHは、乱数ブロックを鍵入力とし、生成された暗号文のメッセージ認証子を生成する。乱数生成は事前計算可能であり、暗号文ブロック生成演算は並列処理可能であり、ハッシュ関数NHも並列処理可能であるため、高速計算ができる。

【0017】

【発明の実施の形態】

（第一の実施形態）

以下、本発明の第一の実施形態を図を用いて説明する。なお、ビットごとの排他的論理和は、以下の説明ではEORと表し、各図においてはプラス記号を円で囲んだ記号で表す。

（第一の実施形態）

図1はネットワーク1001によって接続されたコンピュータA1002、コンピュータB1003を含む、コンピュータA1002からコンピュータB1003への暗号通信を目的としたシステム構成を示すものである。コンピュータA1002は内部に演算装置(以下CPUという)1004、記憶装置(揮発性、不揮発性を問わない。以下RAMという)1005、ネットワークインターフェース1006を装備しており、外部にはコンピュータA1002をユーザが操作するためのディスプレイ1007とキーボード1008が接続されている。RAM1005には暗号化処理プログラムPROG1_1009、乱数生成処理プログラムPROG2_1010、コンピュータA1002とコンピュータB1003間のみで共有されている秘密情報である秘密鍵K1011、コンピュータA1002とコンピュータB1003間で共有さ

れているデータである初期ベクトルI1013, それに暗号化してコンピュータB1003に送信したいデータであるメッセージM1014が保存されている。コンピュータB1003は内部にCPU1015, RAM1016, ネットワークインターフェース1017を装備しており, 外部にはコンピュータB1003をユーザが操作するためのディスプレイ1018とキーボード1019が接続されている。RAM1016には復号化処理プログラムPROG3_1020, 乱数生成処理プログラムPROG2_1021, 秘密鍵K1011が保存されている。

【 0 0 1 8 】

コンピュータA1002は, 暗号化処理プログラムPROG1_1009を実行し, メッセージM1014の暗号文C1022を作成し, ネットワークインターフェース1006を通してネットワーク1001へ送信する。コンピュータB1003は, ネットワークインターフェース1017を通して受信したあと, 復号化処理プログラムPROG3_1020を実行し, もし改ざんが検出されなければ復号化結果をRAM1016に保存する。

【 0 0 1 9 】

各プログラムは, 互いのコンピュータまたは他のコンピュータから通信媒体、すなわちネットワーク1001またはネットワーク1001上を伝搬する搬送波、を介して, またはCD, FDなど可搬型記憶媒体を介してRAMに導入することができる。各プログラムは, 各コンピュータのオペレーティングシステム(図示していない)の元で動作するように構成することも可能である。また、各プログラムによる処理は、CPUが当該プログラムをメモリから読み出して実行することにより、各コンピュータ上で実現される。

【 0 0 2 0 】

暗号化処理プログラムPROG1_1009は, コンピュータA1002において, RAM1005から読み出されて, CPU1004により実行される。暗号化処理プログラムPROG1_1009は, サブルーチンとして乱数生成処理PROG2_1010を内部で呼び出し, 入力秘密鍵K1011, メッセージM1014に対して, 暗号文C1022を出力する。

【 0 0 2 1 】

復号化処理プログラムPROG3_1020は, コンピュータB1003において, RAM1016から読み出されてCPU1015により実行される。復号化処理プログラムPROG3_1020は, サブルーチンとして乱数生成処理PROG2_1021を内部で呼び出し, 入力秘密鍵K1

011, 暗号文C1022に対して, メッセージ, または改ざん検知警告を出力する。

【 0 0 2 2 】

暗号化処理プログラムPROG1_1009の処理の流れを説明する。

ステップ2002: データセットサブルーチン。秘密鍵Kの入力を待つ。

ステップ2003: 平文準備サブルーチン。平文の入力を待ち, 平文が与えられたあと決められたパディングを行い, 最後に64ビットごとに区切って平文ブロックの列 P_i ($1 \leq i \leq N$)を出力する。但し, Nは偶数とする。

ステップ2004: 乱数生成サブルーチン。秘密鍵Kと初期ベクトルIから擬似乱数列 R_i ($1 \leq i \leq N+1$)を出力する。

ステップ2005: 暗号化サブルーチン。擬似乱数列 R_i と平文ブロック列 P_i ($1 \leq i \leq N$)を使って, 暗号文ブロック C_i ($1 \leq i \leq N+2$)を出力する。

ステップ2006: ステップ2005で得られた暗号文ブロック C_i ($1 \leq i \leq N+2$)を順にビット連結し, 暗号文Cとして出力する。

【 0 0 2 3 】

平文準備サブルーチンの処理を図2を用いて説明する。

ステップ2202: 暗号処理に用いるメッセージMの入力を待つ。メッセージMは, キーボード1008から入力されたり, RAM内に保存されていたり, 他の記憶媒体から導入されたりする。

ステップ2203: メッセージの長さを示すデータをパディング。メッセージMの先頭にメッセージMのビット長を表す64ビット二進数データを付加する。

ステップ2204: メッセージ長をそろえるパディング。後の暗号処理のためパディング後のデータを128ビットの整数倍にする。メッセージMの長さをLビットとすると $128 - (L \bmod 128)$ 個の0をステップ2203で長さデータを付加されたメッセージの末尾にパディングする。

ステップ2206: メッセージデータの平文ブロックへ分割。ステップ2205の結果得られたデータを64ビットのブロックに区切り順に P_1, P_2, \dots, P_N とする。

【 0 0 2 4 】

乱数生成サブルーチンの処理を図3を用いて説明する。

ステップ2302: 必要パラメータの入力。パディング後のメッセージブロック数N

と、初期ベクトルI

秘密鍵Kを得る。

ステップ2303：擬似乱数列Rの生成。乱数生成処理プログラムPROG2を呼び出し、長さ64(N+1)ビットの擬似乱数列を生成、出力をRとする。

ステップ2304：乱数列Rをブロックに分割。擬似乱数列Rを64ビットごとに区切り順に R_1, R_2, \dots, R_{N+1} とする。

【 0 0 2 5 】

暗号化、メッセージ認証子生成セットアップのサブルーチンの処理を図4を用いて説明する。

ステップ2403：カウンタ初期化。 $i=1$ とする。

ステップ2404：暗号文ブロック C_i 計算。 $C_i \leftarrow M_i \text{ EOR } R_i$ とする。

ステップ2406： $i=N$ ならばステップ2408を実行。

ステップ2407：カウンタ i をインクリメントしステップ2404へ戻る。

ステップ2408： $C_i (1 \leq i \leq N)$ を順にビット連結し、Sとする。 $R_i (2 \leq i \leq N+1)$ を順にビット連結し、Rとする。

ステップ2409： $NH_R(S)$ の出力を64ビットごとに区切り、 C_{N+1}, C_{N+2} とする。

【 0 0 2 6 】

ハッシュ関数 $NH_R()$ については、図11で解説する。

【 0 0 2 7 】

復号化処理プログラムPROG3_1020の処理の流れを図5を用いて説明する。

ステップ2502：データセットサブルーチン。秘密鍵Kの入力を待つ。

ステップ2503：暗号文準備サブルーチン。暗号文 C' の入力を待ち、暗号文 C' が与えられたあと64ビットごとに区切って暗号文ブロックの列 $C'_i (1 \leq i \leq N+2)$ を出力する。

ステップ2504：乱数生成サブルーチン。秘密鍵Kから擬似乱数列 $R_i (1 \leq i \leq N+1)$ を出力する。

ステップ2505： $C_i (1 \leq i \leq N)$ を順にビット連結し、Sとする。 $R_i (2 \leq i \leq N+1)$ を順にビット連結し、Rとして $NH_R(S)$ を計算する。

ステップ2506： $NH_R(S) = C'_{N+1} || C'_{N+2}$ ならばステップ2508に進む。そうで

ないならばステップ2507に進む。

ステップ2507：拒否(非受理)を出力。ステップ2511に進む。

ステップ2508：復号化サブルーチン。擬似乱数列 R_i ($1 \leq i \leq N$)，暗号文ブロック列 C'_i ($1 \leq i \leq N$)を使って，平文ブロック P'_i ($1 \leq i \leq N$)を出力する。

ステップ2509：平文切り出しサブルーチン。平文ブロックの列 P'_i をデータ列， L' ， M' に分割する。

ステップ2510： M' をRAMへ格納する。

ステップ2511では，復号化処理プログラムは，結果(受理/非受理あるいは復号結果)をディスプレイ1018に出力して，ユーザに結果を通知する。

【 0 0 2 8 】

暗号文準備サブルーチンの処理を図6を用いて説明する。

ステップ2602：暗号文 C' の入力を待つ。

ステップ2603：暗号文 C' を64ビットごとに区切り，順に C'_1 ， C'_2 ， \dots ， C'_N ， C'_{N+1} ， C'_{N+2} とする。

【 0 0 2 9 】

復号化サブルーチンの処理を図7を用いて説明する。

ステップ2703：カウンタ初期化。 $i=1$ とする。

ステップ2704：平文ブロック P'_i 計算。 $P'_i = C'_i \wedge R_i$ とする。

ステップ2706： $i=N$ でないならばステップ2707を実行。

ステップ2707：カウンタ i をインクリメントしステップ2704へ戻る。

【 0 0 3 0 】

平文切り出しサブルーチンの処理を図8を用いて説明する。

ステップ2802： L' を最初の64ビットの平文ブロックとする。

ステップ2803：復号文ブロックのうち P'_2 の最上位ビットから L' ビットのデータまでを M' とする。

【 0 0 3 1 】

図9は，暗号化処理の説明図である。

【 0 0 3 2 】

メッセージM2931に長さ2930，適当なパディング2932をそれぞれ付加し，平文P

2934を生成する。

【 0 0 3 3 】

これを64ビットに分割したものをそれぞれ P_{1_2935} , P_{2_2936} , ..., P_{N_2938} とする。

【 0 0 3 4 】

P_{1_2935} は R_{1_2920} と排他論理和を取り、暗号文ブロック C_{1_2943} を得る。

【 0 0 3 5 】

P_{2_2936} は R_{2_2921} と排他論理和を取り、暗号文ブロック C_{2_2944} を得る。

【 0 0 3 6 】

同様にして P_{N_2938} まで処理を行い、暗号文ブロック C_{1_2943} , C_{2_2944} , ..., C_{N_2947} を得る。 R_{2_2921} , R_{3_2921} , R_{N+1_2928} をこの順で連結したものと C_{1_2943} , C_{2_2944} , ..., C_{N_2947} をこの順で連結したものを S を入力として $NH_R(S)$ を計算する。

$NH_R(S)$ の出力を C_{N+1_2948} , C_{N+2_2949} とブロック分割し、 C_{1_2943} , C_{2_2944} , ..., C_{N_2947} , C_{N+1_2948} , C_{N+2_2949} この順で連結し、暗号文 C_{2956} を得る。

【 0 0 3 7 】

図10は、復号化処理の説明図である。

【 0 0 3 8 】

暗号文 $C'_{_4030}$ を64ビットのブロックに分割し、 C'_{1_4035} , C'_{2_4036} , ..., C'_{N_7037} , C'_{N+1_4038} , C'_{N+2_4039} とする。

R_{2_4021} , R_{3_4022} , ..., R_{N+1_4028} をこの順で連結したものと C'_{1_4035} , C'_{2_4036} , ..., C'_{N_7037} をこの順で連結したものを S を入力として $NH_R(S)$ を計算し、 $NH_R(S) = C_{N+2'}_{_4038} || C_{N+1'}_{_4039}$ ならば、次に進む。

【 0 0 3 9 】

$C_{1'}_{_4035}$ は R_{1_4020} と排他論理和を取り、平文ブロック $P_{1'}_{_4043}$ を得る。

【 0 0 4 0 】

$C_{2'}_{_4036}$ は R_{2_4021} と排他論理和を取り、平文ブロック $P_{2'}_{_4044}$ を得る。

【 0 0 4 1 】

同様にして $C_{N'}_{_4037}$ まで処理を行い、平文ブロック $P_{1'}_{_4043}$, $P_{2'}_{_4044}$,

..., P_N' _4047を得た後, これらをこの順に連結し, 平文 P' _4050とする。これを L' _4051, M' _4052に分割する。

【0 0 4 2】

図11を用いて, 非特許文献2で扱っているハッシュ関数 $NH_R(S)$ の解説を行う。

【0 0 4 3】

この関数はメッセージ M と鍵 K を入力とし, メッセージ認証子 C を生成し, 出力する。は次のようにして行う。ただし, 以下のアルゴリズムで, 矢印 \leftarrow はデータの代入を, $||$ は結合を, それぞれ表す。 $M=M_1||\cdots|| M_N$, $K=K_1||\cdots|| K_N$ とする。

【0 0 4 4】

【数1】

$$\begin{aligned} H_i &\leftarrow M_i + K_i \quad (1 \leq i \leq N) \\ S_i &\leftarrow H_{2i-1} \times H_{2i} \quad (1 \leq i \leq N/2) \\ C &\leftarrow S_1 + S_2 \dots + S_{N/2} \end{aligned}$$

【0 0 4 5】

最後にメッセージ認証子 C を出力する。

【0 0 4 6】

第一の実施形態において, 暗号処理とメッセージ認証子生成という二つの処理に必要な擬似乱数の長さはメッセージのそれとほぼ同じで十分である。

【0 0 4 7】

また, 一般的なCPUを使用した計算機上で, 本実施形態による擬似乱数生成器はブロック暗号の中で最も高速なAESに比べ2倍以上高速な処理が可能である。従って, 従来技術であるiaPCBCモードに比べ, 同一環境上で2倍以上高速な処理が可能である。

(第二の実施形態)

以下, 本発明の第二の実施形態について説明する。基本的には, 第一の実施形態と同じだが, 変更点だけ以下に示す。

【0 0 4 8】

暗号化処理プログラムPROG1_1009の処理の流れを説明する。

ステップ5002：データセットサブルーチン。秘密鍵Kの入力を待つ。

ステップ5003：平文準備サブルーチン。平文の入力を待ち、平文が与えられたあと決められたパディングを行い、最後に64ビットごとに区切って平文ブロックの列 P_i ($1 \leq i \leq N$)を出力する。但し、Nは偶数とする。

ステップ5004：乱数生成サブルーチン。秘密鍵Kと初期ベクトルIから $64(3N/2+1)$ ビットの擬似乱数列を出力する。

ステップ5005：暗号化サブルーチン。ステップ5004で得られた擬似乱数列と平文ブロック列 P_i ($1 \leq i \leq N$)を使って、暗号文ブロック C_i ($1 \leq i \leq N+2$)を出力する。

ステップ5006：ステップ5005で得られた暗号文ブロック C_i ($1 \leq i \leq N+2$)を順にビット連結し、暗号文Cとして出力する。

【 0 0 4 9 】

乱数生成サブルーチンの処理を図12を用いて説明する。

ステップ5302：必要パラメータの入力。パディング後のメッセージブロック数Nと、初期ベクトルIと、秘密鍵Kを得る。

ステップ5303：擬似乱数列Rの生成。乱数生成処理プログラムPROG2を呼び出し、長さ

$64(3N/2+1)$ ビットの擬似乱数列Rを生成する。

ステップ5304：乱数列Rをブロックに分割。擬似乱数列Rを64ビットごとに区切り順に $R_1, R_2, \dots, R_{N+1}, \dots, R_{3N/2+1}$ とする。

ステップ5305： $R_{N+1}, \dots, R_{3N/2}$ をこの順に連結し、 R' とする。

ステップ5306： $R_{N+2}, \dots, R_{3N/2+1}$ をこの順に連結し、 R'' とする。

【 0 0 5 0 】

暗号化、メッセージ認証子生成セットアップのサブルーチンの処理を図13を用いて説明する。

ステップ5403：カウンタ初期化。 $i=1$ とする。

ステップ5404：暗号文ブロック C_i 計算。 $C_i \leftarrow M_i \text{ EOR } R_i$ とする。

ステップ5405： $i=N$ ならばステップ5407を実行。

ステップ5406：カウンタ i をインクリメントしステップ5404へ戻る。

ステップ5407：カウンタ初期化。 $i=1$ とする。

ステップ5408： C_i を32ビットごとに区切り， C_i, H, C_i, L とする。

ステップ5409： $i=N/2$ ならばステップ5411を実行。

ステップ5410：カウンタ i をインクリメントしステップ5408へ戻る。

ステップ5411： $C_1, H, C_1, L, \dots, C_{N/2}, H, C_{N/2}, L$ を順にビット連結し， S とする。

ステップ5412： $NH_{R'}(S)$ の出力を C_{N+1} とする。

ステップ5413： $NH_{R'}, (S)$ の出力を C_{N+2} とする。

【 0 0 5 1 】

復号化処理プログラムの処理の流れを図14を用いて説明する。

ステップ5502：データセットサブルーチン。秘密鍵 K の入力を待つ。

ステップ5503：暗号文準備サブルーチン。暗号文 C' の入力を待ち，暗号文 C' が与えられたあと64ビットごとに区切って暗号文ブロックの列 $C'_i (1 \leq i \leq N+2)$ を出力する。

ステップ5504：乱数生成サブルーチン。秘密鍵 K から擬似乱数列 $R_i (1 \leq i \leq 3N/2 + 1)$ ， R' ， R'' を出力する。

ステップ5505： $C_i (1 \leq i \leq N)$ を順にビット連結し， S とする。 $NH_{R'}(S)$ と $NH_{R'}, (S)$ を計算する。

ステップ5506： $NH_{R'}(S) = C'_{N+1}$ かつ $NH_{R'}, (S) = C'_{N+2}$ ならばステップ5508に進む。そうでないならばステップ5507に進む。

ステップ5507：拒否(非受理)を出力。ステップ5511に進む。

ステップ5508：復号化サブルーチン。擬似乱数列 R_i ，暗号文ブロック列 $C'_i (1 \leq i \leq N)$ を使って，平文ブロック $P'_i (1 \leq i \leq N)$ を出力する。

ステップ5509：平文切り出しサブルーチン。平文ブロックの列 P'_i をデータ列 L' ， M' に分割する。

ステップ5510： M' をRAMへ格納する。

ステップ5510では，復号化処理プログラムは，結果(受理/非受理あるいは復号結果)をディスプレイ10018に出力して，ユーザに結果を通知する。

【 0 0 5 2 】

図15は、暗号化処理の説明図である。

【 0 0 5 3 】

メッセージM5931に長さ5930、適当なパディング5932をそれぞれ付加し、平文P5934を生成する。これを64ビットに分割したものをそれぞれP₁_5935, P₂_5936, P_{N/2}_5937, ..., P_N_5938とする。P₁_5935はR₁_5920と排他論理和を取り、暗号文ブロックC₁_5943を得る。P₂_5936はR₂_5921と排他論理和を取り、暗号文ブロックC₂_5944を得る。

【 0 0 5 4 】

同様にしてP_N_5938まで処理を行い、暗号文ブロックC₁_5943, C₂_5944, ..., C_{N/2}_5947を得る。C₁_5943, C₂_5944, ..., C_{N/2}_5945をこの順で連結したものSを入力としてNH_{R'}(S)を計算し、その出力をC_{N+1}_5948とする。

【 0 0 5 5 】

NH_{R'}(S)を計算し、その出力をC_{N+2}_5949とする。C₁_5943, C₂_5944, ..., C_{N/2}_5945, ..., C_N_5947, C_{N+1}_5948, C_{N+2}_5949この順で連結し、暗号文C_5956を得る。

【 0 0 5 6 】

図16は、復号化処理の説明図である。

暗号文C'_6030を64ビットのブロックに分割し、C'_1_6035, C'_2_6036, ..., C'_N_7037, C'_{N+1}_6038, C'_{N+2}_6039とする。C'_1_6033, C'_2_6034, C'_{N/2}_7035, ..., C'_N_7037をこの順で連結したものSを入力としてNH_R(S)を計算し、NH_{R'}(S)=C_{N+1}'_6038かつ、NH_{R'}(S)=C_{N+2}'_6039ならば、次に進む。

【 0 0 5 7 】

C'_1_6033はR₁_6020と排他論理和を取り、平文ブロックP'_1_6043を得る。C'_2_6034はR₂_6031と排他論理和を取り、平文ブロックP'_2_6044を得る。

【 0 0 5 8 】

同様にしてC'_N_6037まで処理を行い、平文ブロックP'_1_6043, P'_2_6044, ..., P'_N_6047を得た後、これらをこの順に連結し、平文P'_6050とする。これをL'_6051, M'_6052に分割する。

【0 0 5 9】

第二の実施形態において、暗号処理とメッセージ認証子生成という二つの処理に必要な擬似乱数の長さはメッセージのそれのほぼ1.5倍である。また、一般的なCPUを使用した計算機上で、本実施形態による疑似乱数生成器はブロック暗号の中で最も高速なAESに比べ2倍以上高速に乱数生成処理が可能である。以上の考察から、第二の実施形態の方法は、従来技術であるiaPCBCモードに比べ、同一環境上で、4/3倍以上高速な処理が可能である。

【0 0 6 0】

また、非特許文献2の定理2を、 $w = 32$ ， $t = 2$ として第二の実施形態に適用することにより、安全性証明ができる。すなわち、長さが同じである二つの異なるメッセージに対し、それらのメッセージ認証子が等しくなる確率は 2^{-64} である。

【0 0 6 1】

【発明の効果】

本発明によれば、メッセージ認証付き暗号方法をソフトウェアで実装する際、処理速度を高速化することができる。

【図面の簡単な説明】

【図 1】

各実施形態のシステム構成図を示す。

【図 2】

平文準備サブルーチンのフロー図を示す。

【図 3】

乱数生成サブルーチンのフロー図を示す。

【図 4】

暗号化サブルーチンのフロー図を示す。

【図 5】

図1の復号化処理プログラムのフロー図を示す。

【図 6】

暗号文準備サブルーチンのフロー図を示す。

【図 7】

復号化サブルーチンのフロー図を示す。

【図 8】

平文切出しサブルーチンのフロー図を示す。

【図 9】

暗号化処理の、データブロックによる図を示す。

【図 1 0】

復号化処理の、データブロックによる図を示す。

【図 1 1】

ハッシュ関数NHのフロー図を示す。

【図 1 2】

第二の実施形態の乱数生成2サブルーチンのフロー図を示す。

【図 1 3】

第二の実施形態の暗号化2サブルーチンのフロー図を示す。

【図 1 4】

第二の実施形態の復号化処理プログラムのフロー図を示す。

【図 1 5】

第二の実施形態の暗号化処理の、データブロックによる図を示す。

【図 1 6】

第二の実施形態の復号化処理の、データブロックによる図を示す。

【図 1 7】

第一の実施形態の暗号化処理と認証処理における、乱数共有方法の概念図を示す。

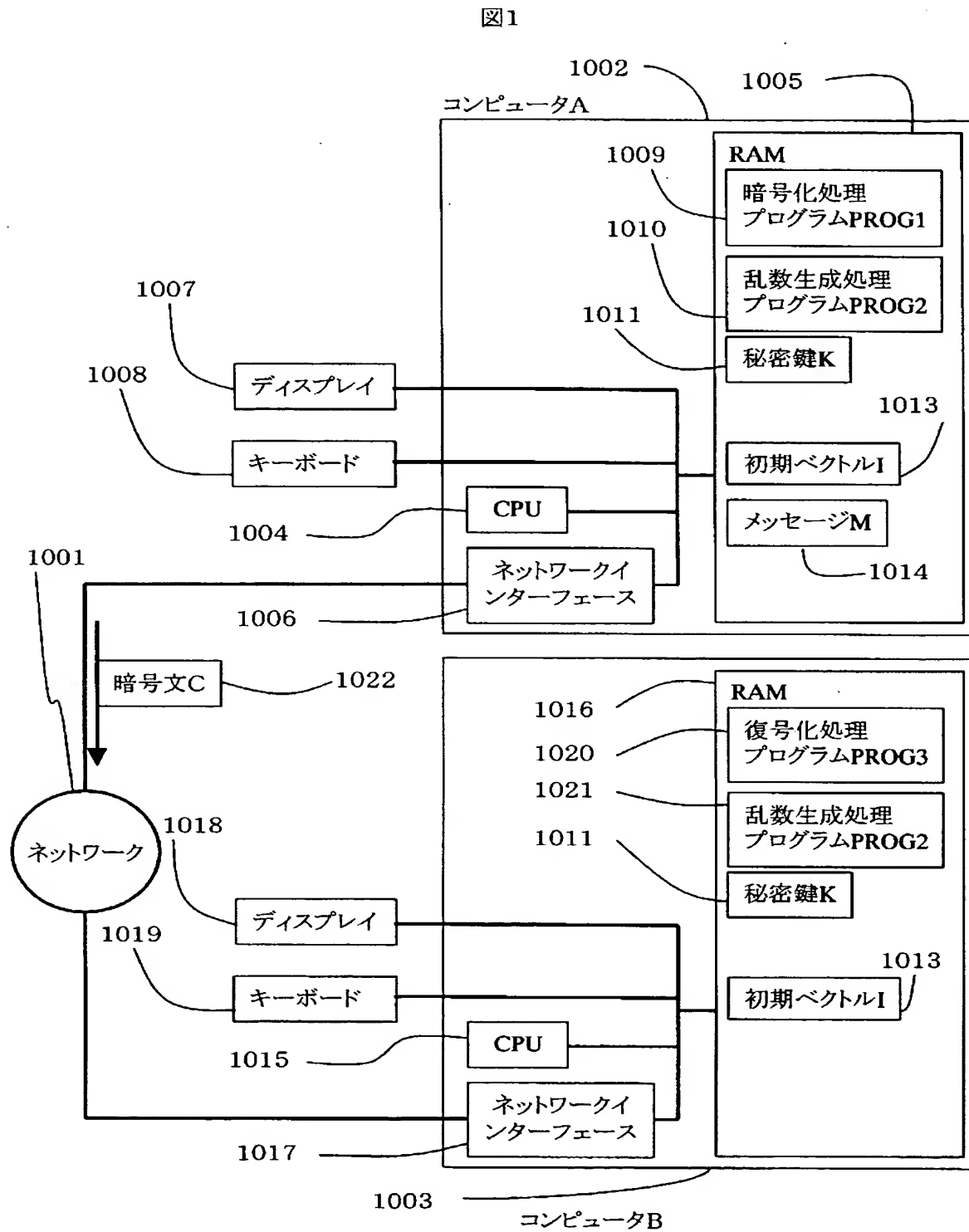
【符号の説明】

1001…ネットワーク、1002…コンピュータA、1003…コンピュータB、1004…CPU、1005…RAM、1006…ネットワークインターフェース、1007…ディスプレイ、1008…キーボード、1009…暗号化処理プログラムPROG1_、1010…乱数生成処理プログラムPROG2_、1011…秘密鍵K、1013…初期ベクトルI、1014…メッセージM、1015…CPU、1016…RAM、1017…ネットワークインターフェース、1018…ディスプレイ、1019…キーボード、1020…復号化処理プログラムPROG3_、1021…乱数生成処

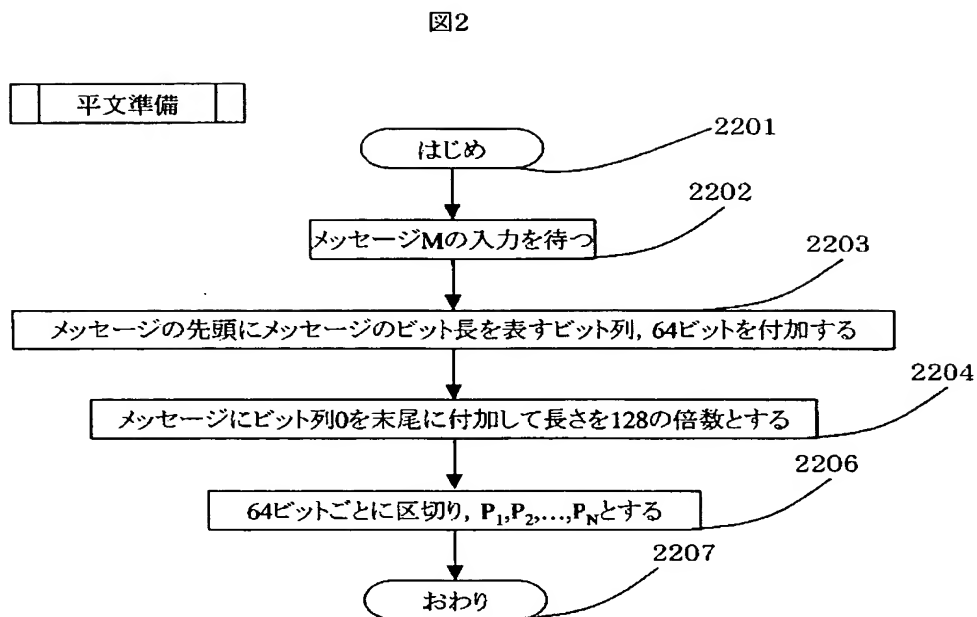
理プログラムPROG2_、1022…暗号文C。

【書類名】 図面

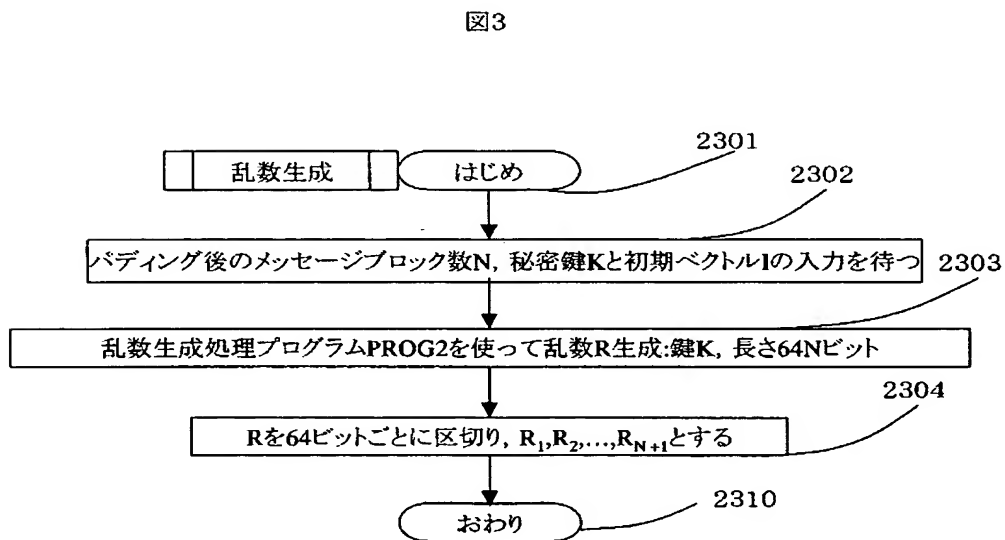
【図 1】



【図 2】

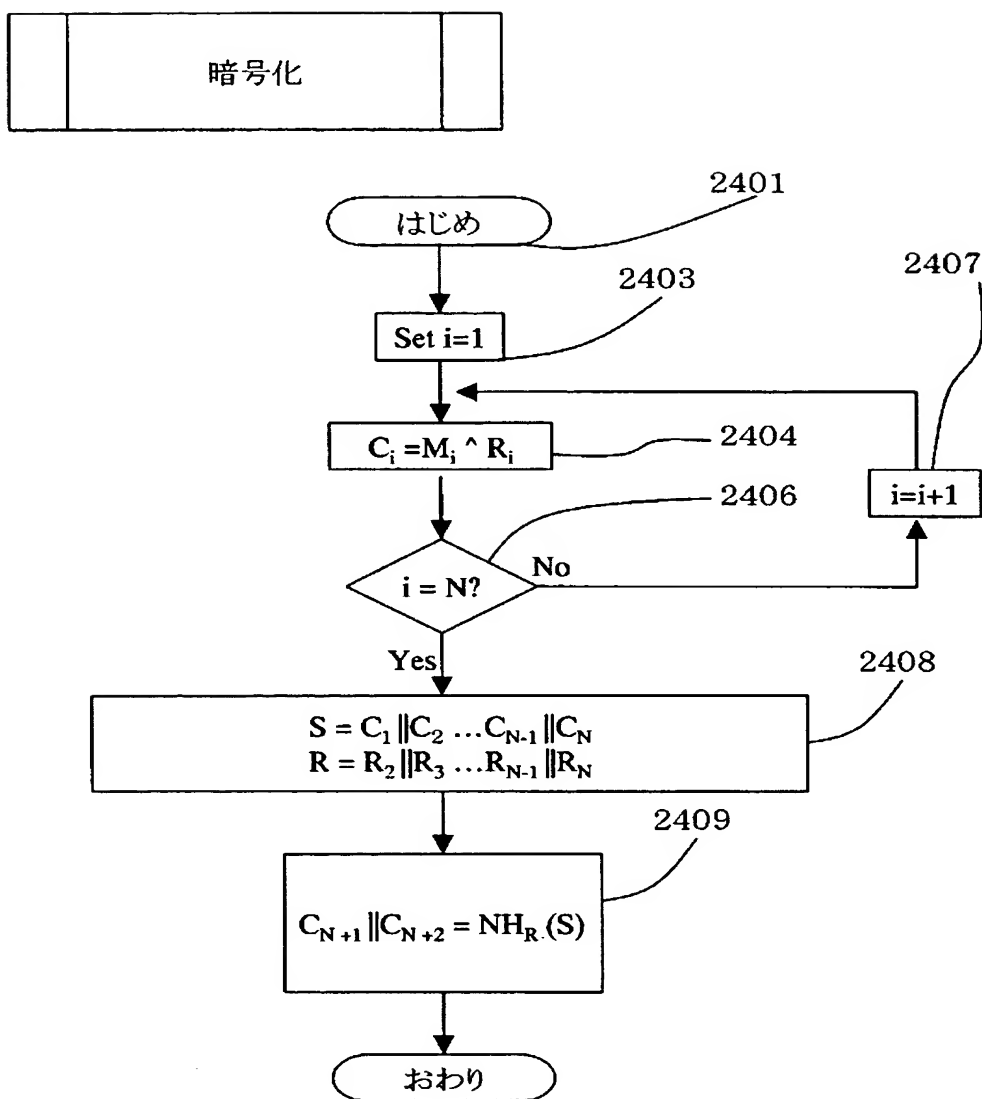


【図 3】



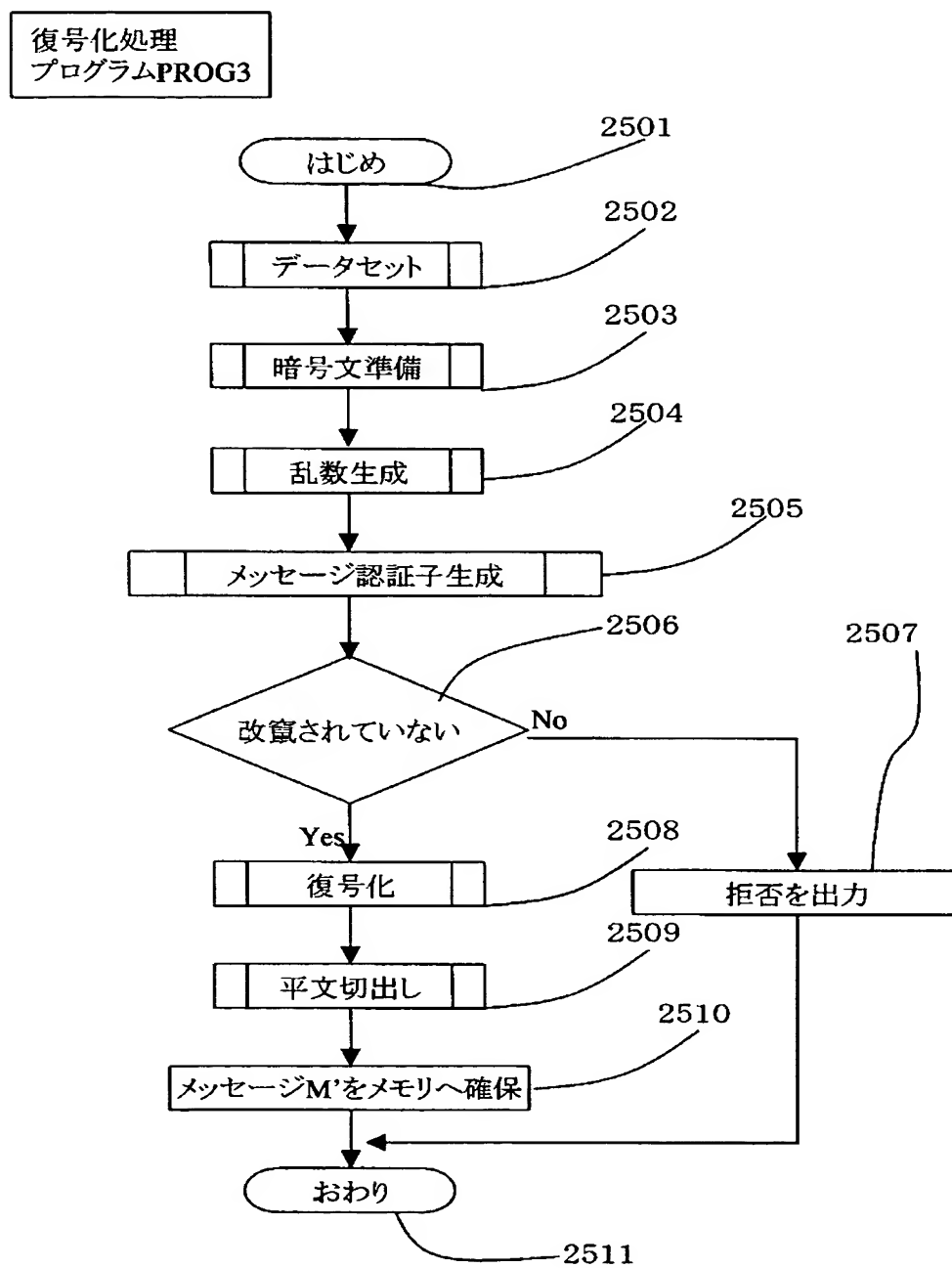
【図 4】

図4



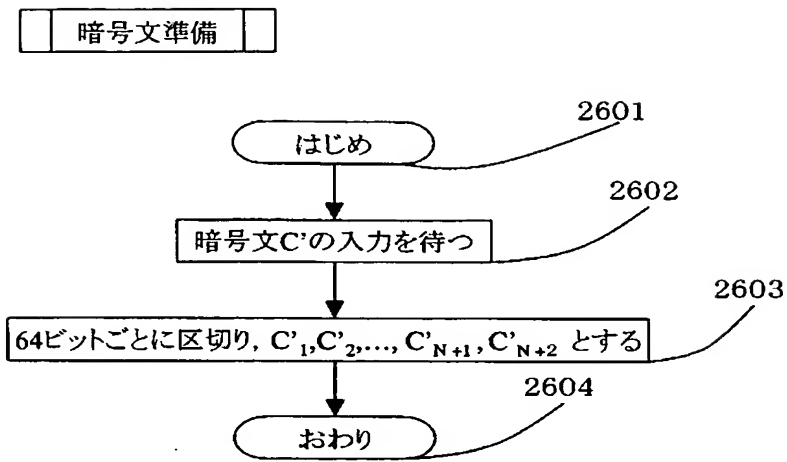
【図 5】

図5



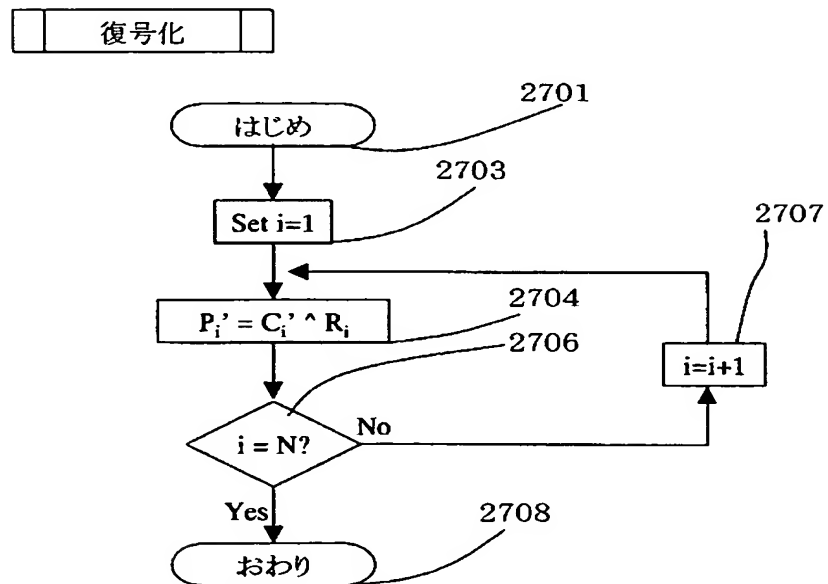
【図 6】

図6



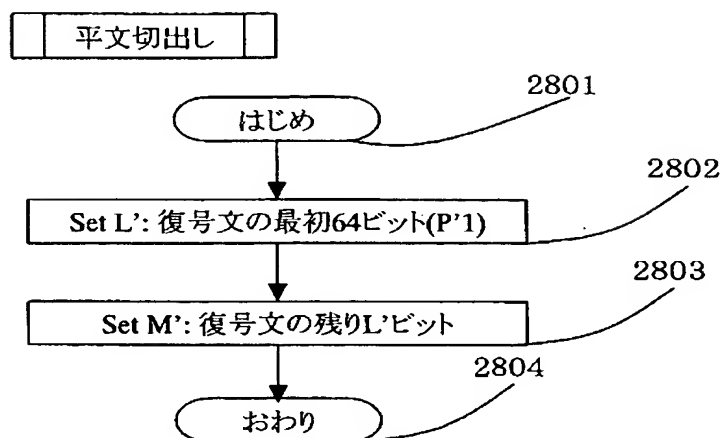
【図 7】

図7



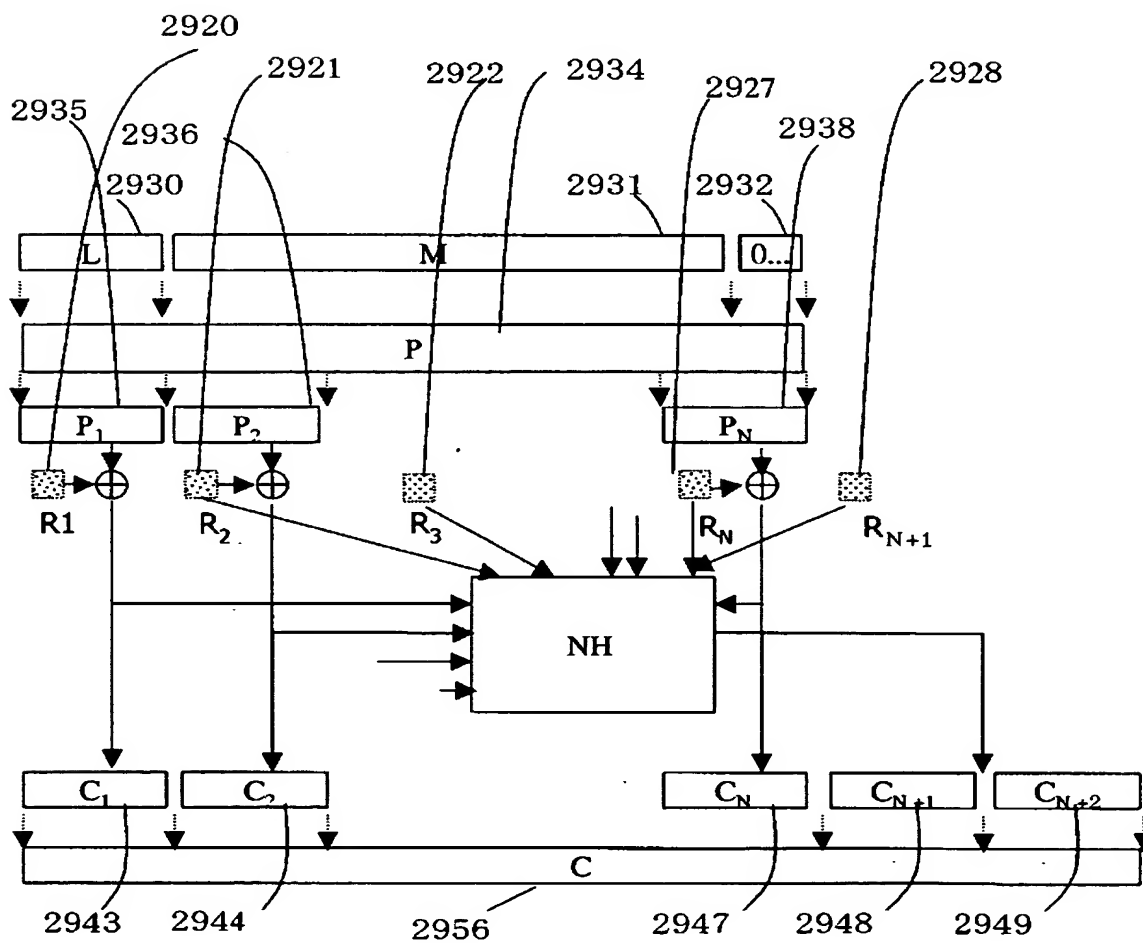
【図 8】

図8

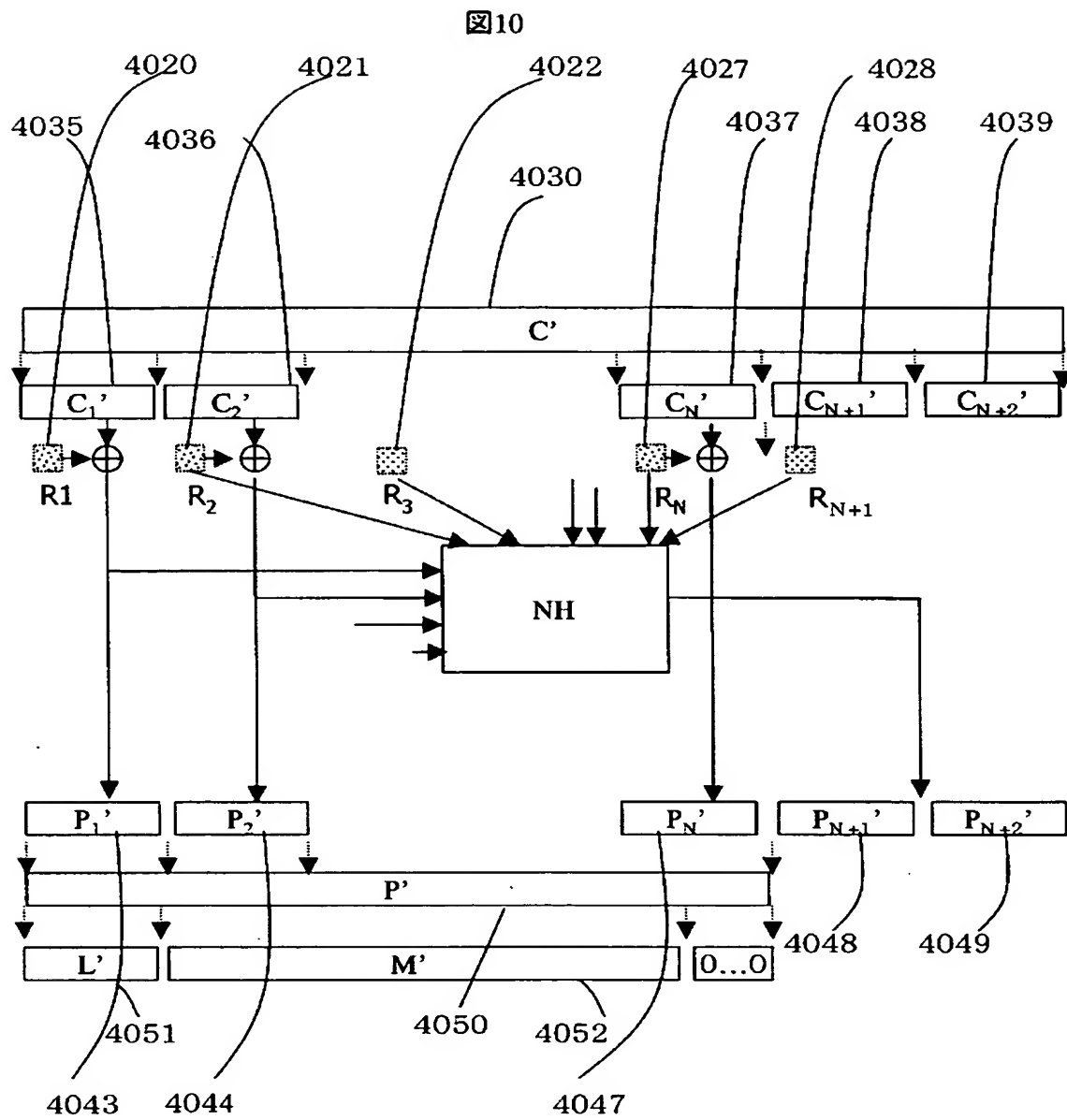


【図 9】

図9

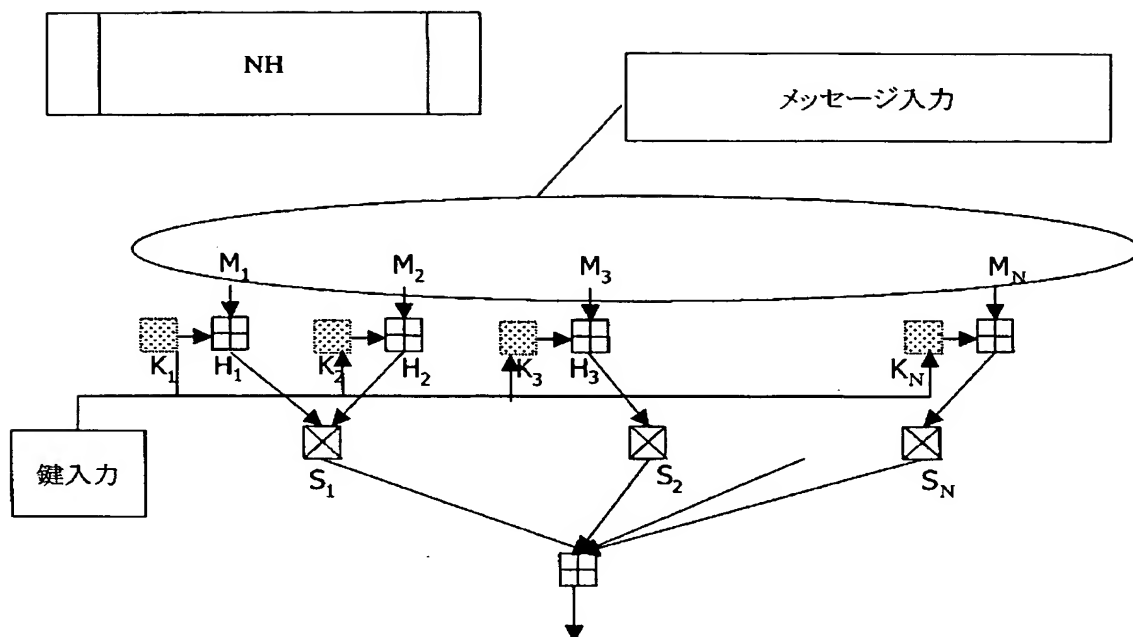


【図 10】



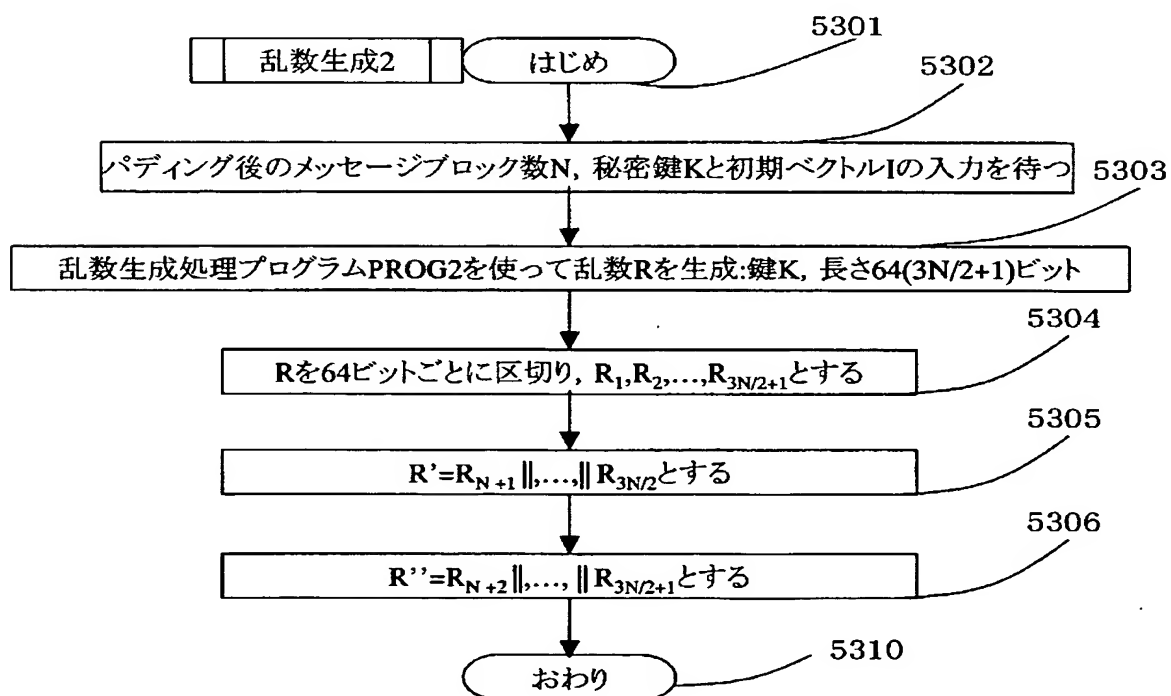
【図 11】

図11

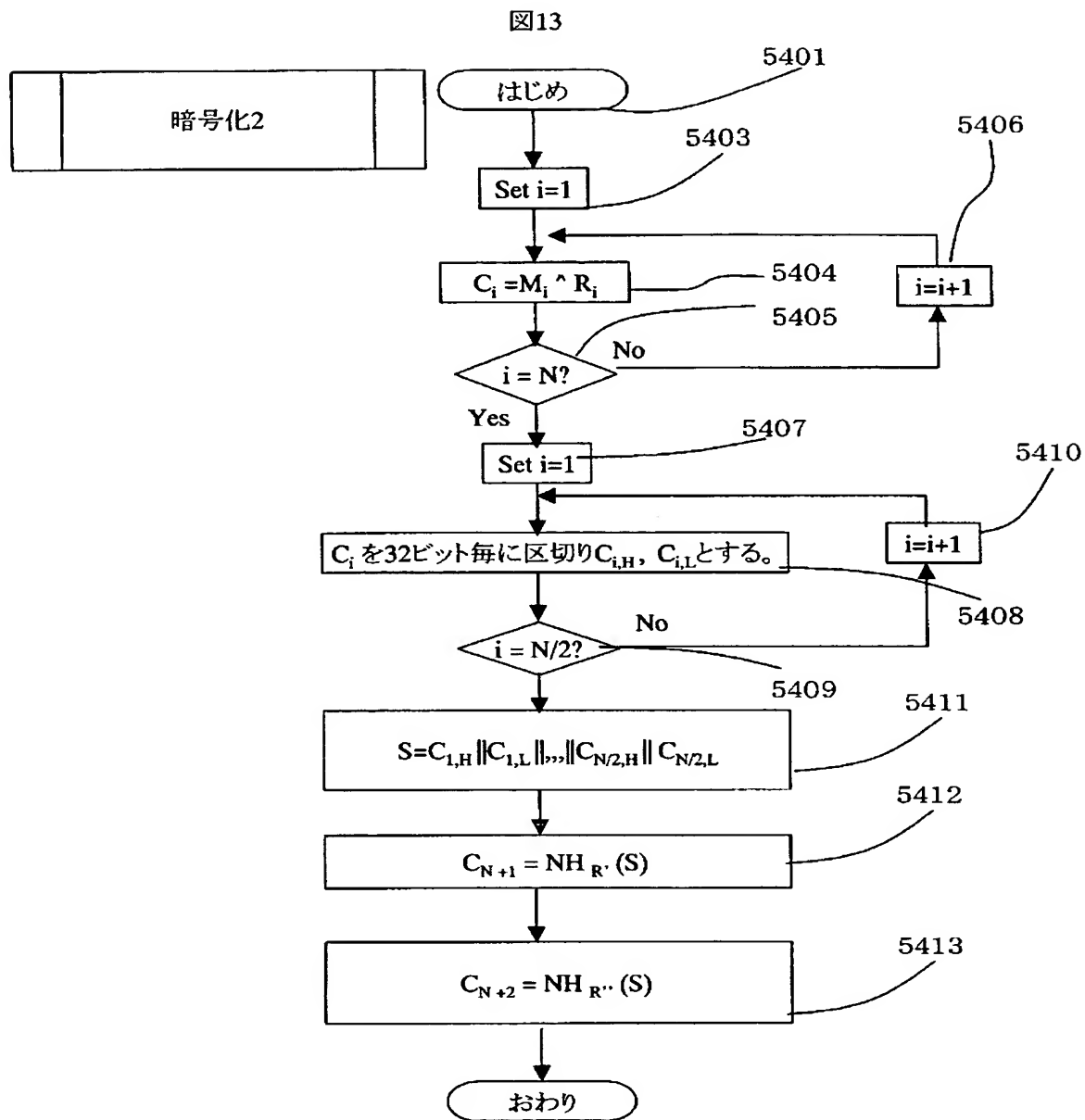


【図 12】

図12

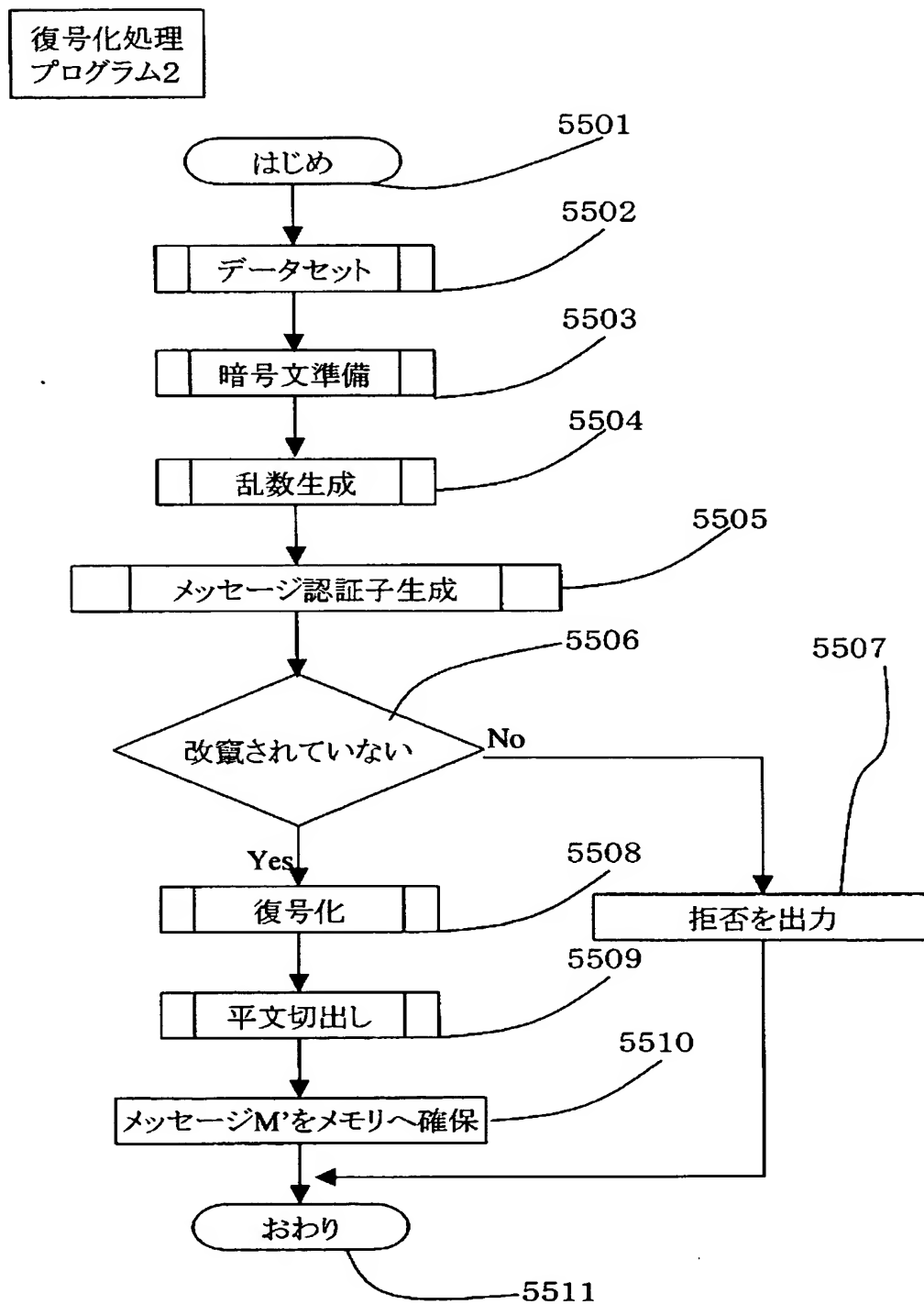


【図 13】

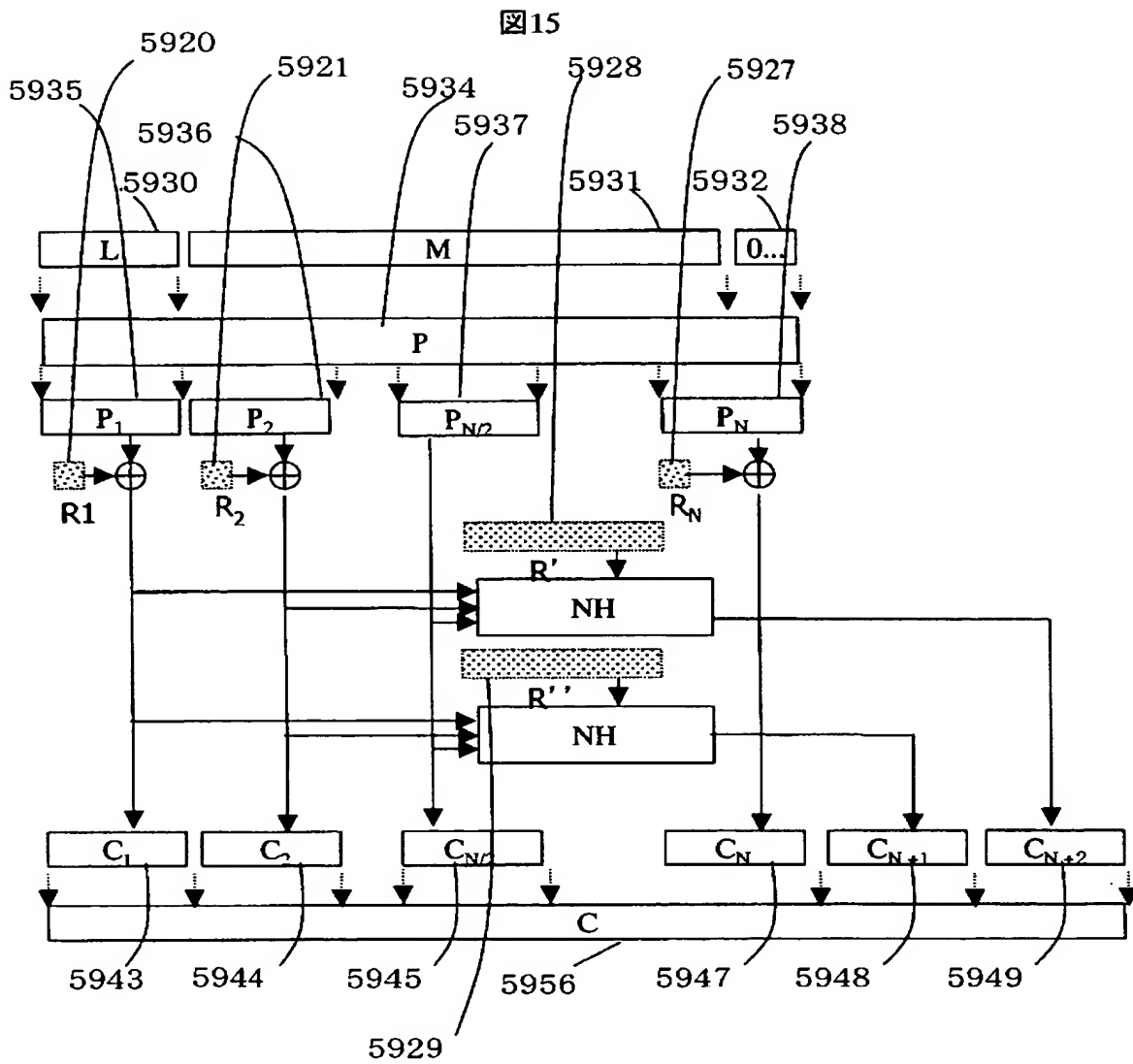


【図14】

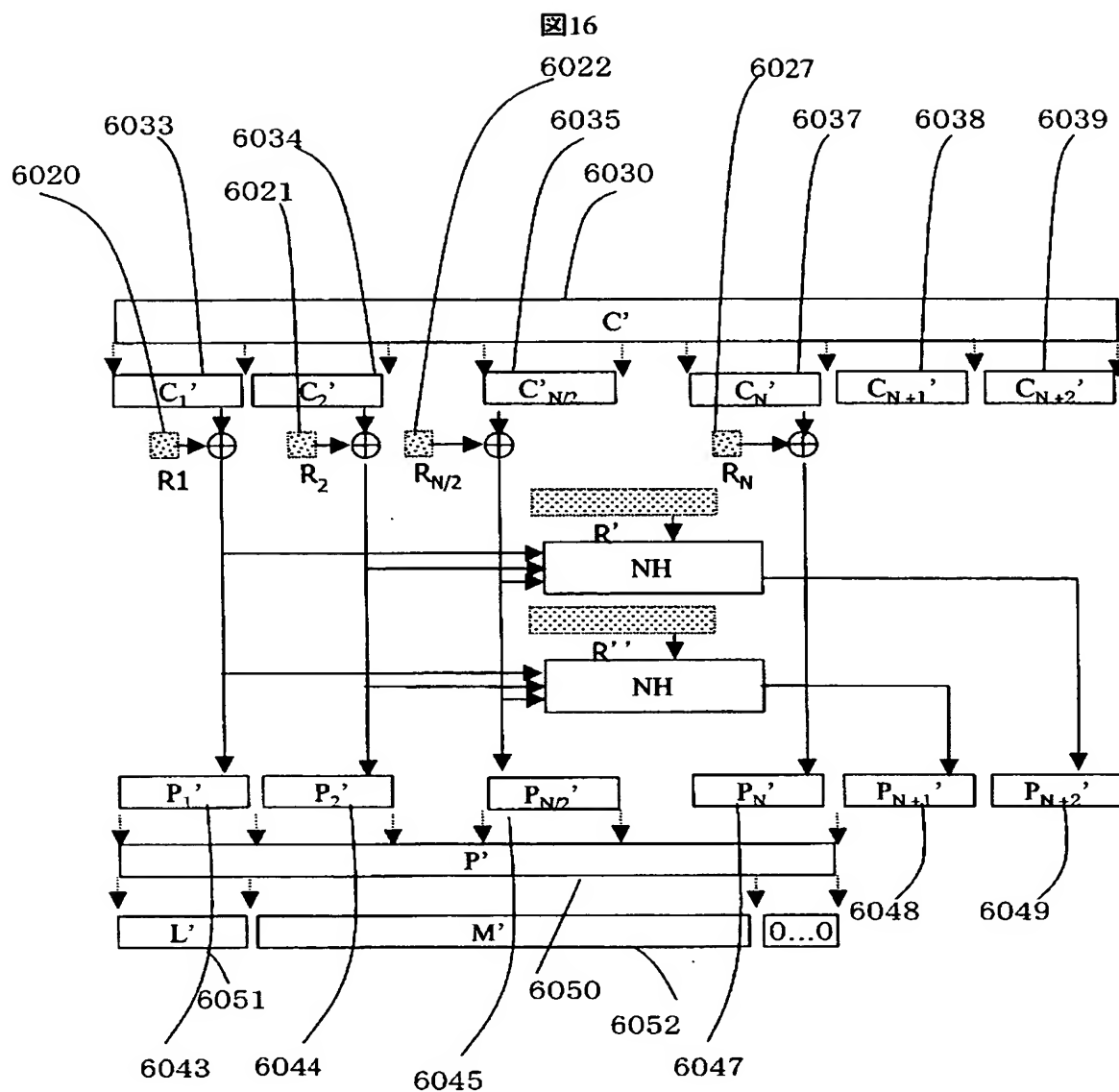
図14



【図 15】

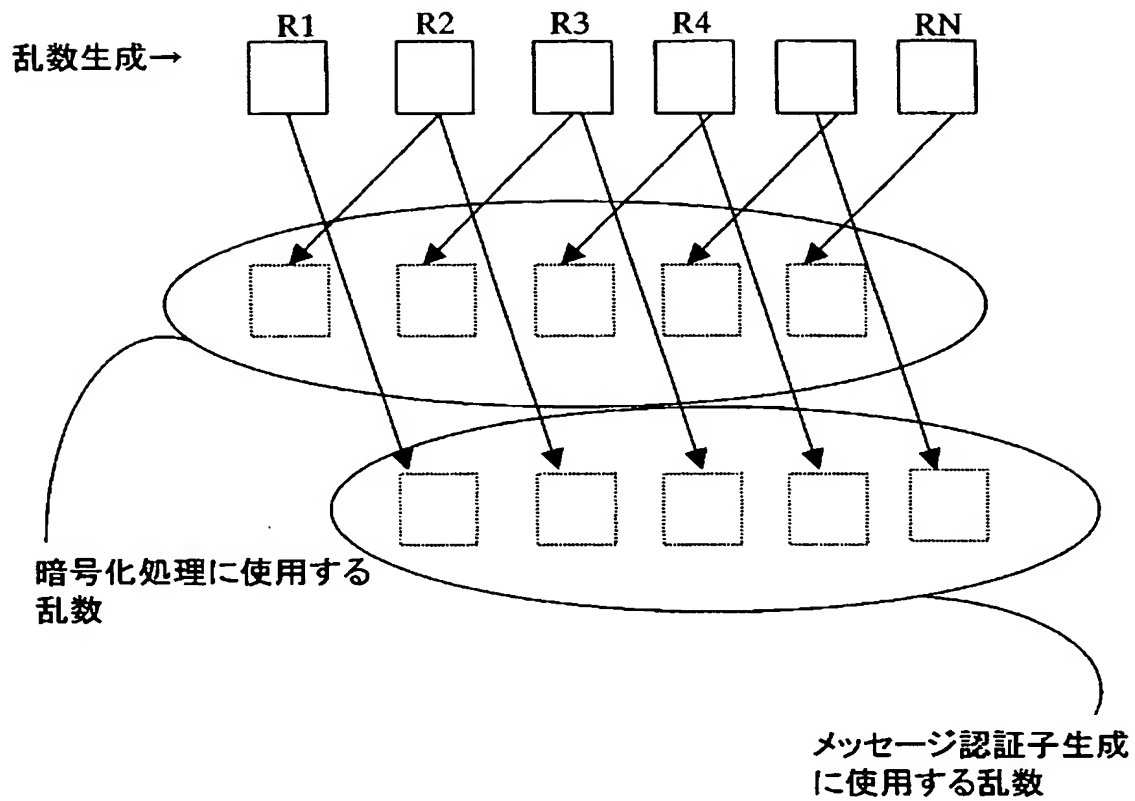


【図 16】



【図 17】

図17



【書類名】 要約書**【要約】****【課題】**

従来の暗号技術では、復号の際、改ざん検出を行おうとする、異なるふたつの鍵共有の必要性、メッセージの2倍となる乱数の必要性、独立の処理、別の暗号学的要素関数の追加実装などが必要であった。

【解決手段】

乱数を生成して暗号化と認証処理を行い、事前計算と並列計算を達成する。また、生成する乱数の長さは、メッセージ長 N に対して $2N$ より少ない乱数を用いて、暗号処理と認証処理を行う。具体的には、疑似乱数生成器を使って乱数を生成しそれらをブロック毎に分割する。また平文もブロック毎に分割する。乱数ブロックと平文ブロックを排他的論理和し暗号文ブロックを得る。ハッシュ関数 NH は、乱数ブロックを鍵入力とし、生成された暗号文のメッセージ認証子を生成する。乱数生成は事前計算可能であり、暗号文ブロック生成演算は並列処理可能であり、ハッシュ関数 NH も並列処理可能であるため、高速計算ができる。

【選択図】 図1

認定・付加情報

特許出願の番号	特願 2 0 0 3 - 1 5 7 4 4 4
受付番号	5 0 3 0 0 9 2 1 7 6 1
書類名	特許願
担当官	第七担当上席 0 0 9 6
作成日	平成 1 5 年 6 月 4 日

< 認定情報・付加情報 >

【提出日】	平成15年 6月 3日
-------	-------------

次頁無

特願 2 0 0 3 - 1 5 7 4 4 4

出 願 人 履 歴 情 報

識別番号 [0 0 0 0 0 5 1 0 8]

1. 変更年月日	1 9 9 0 年 8 月 3 1 日
[変更理由]	新規登録
住 所	東京都千代田区神田駿河台 4 丁目 6 番地
氏 名	株式会社日立製作所